

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Zobrazení fázorového diagramu
přenosové sítě na PC**

Presentation of Transmission Network Phasors
Diagram on PC

Zadání bakalářské práce

Student: **Lukáš Nevřela**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Zobrazení fázorového diagramu přenosové sítě na PC**
Presentation of Transmission Network Phasors Diagram on PC

Zásady pro vypracování:

Vytvořte přímo spustitelnou aplikaci pro zobrazování fázorových diagramů přenosové sítě na počítači. Vstupními prvky budou jednotlivé vektory proudů a napětí s vazbou na typ zapojení sítě. Výsledkem má být fázorový diagram napětí a proudů. Pro vytvoření aplikace použijte např. technologii Adobe Flash, Silverlight, Java Applet, apod. V teoretické části práce zpracujte typy přenosových sítí z hlediska počtu fází, typů zapojení včetně respektování impedancí vedení. Zpracujte a popište algoritmy použité pro zadávání, přepočty a zobrazování vektorů. Vytvořte samospustitelné instalační CD. Práci zpracujte dle následujících bodů:

1. Návrh možností vykreslení fázorových diagramů napětí a proudů pro střídavé obvody včetně třífázových.
2. Návrh způsobu zadávání vektorů.
3. Vykreslení fázorového diagramu v polárních nebo kartézských souřadnicích.
4. Zhodnocení předností a nedostatků při různých způsobech zadávání vektorů, zobrazení, případně pro různé programy.
5. Vytvořte uživatelský návod na použití vytvořeného programu

Seznam doporučené odborné literatury:

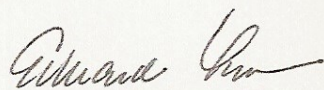
Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

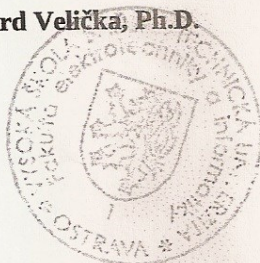
Vedoucí bakalářské práce: **Ing. Richard Velička, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7. 5. 2014

Lukáš Váňa

Podpis

Poděkování

Na tomto místě bych rád poděkoval Ing. Richardu Veličkovi, Ph.D. za jeho cenné rady a připomínky a vedení při vypracování této práce. Dále bych rád poděkoval Ing. Petru Orságovi, Ph.D. za jeho pomoc a poskytnuté konzultace.

Abstrakt

Tato práce se zabývá vytvořením programu, který bude sloužit k výpočtu a zobrazení fázorového diagramu pro trojfázovou soustavu. Program bude obsahovat grafické rozhraní pro interakci s uživatelem. V tomto rozhraní bude uživatel zadávat hodnoty jednotlivých prvků, výběr typu zapojení, výběr zobrazovaných prvků a možnost pootočení grafu o určitý úhel. Výstupem tohoto programu bude zobrazený diagram a okno obsahující vypočtené výsledky.

Klíčová slova

Trojfázová soustava, fázor, fázorový diagram, fáze, proud, napětí, zdroj, zátěž, vedení, impedance, admittance, odpor, indukčnost, kapacita, frekvence, hvězda, trojúhelník, Java, Swing.

Abstract

The goal of this work is to create an application which will be used to calculate and render phasor diagrams for a three-phase system. The application will include a graphical interface to interact with the user. In this interface, the user can enter values of individual elements, select the type of connection and select displayed elements. The interface also gives the user the ability to rotate graphs by a certain angle. The output of this program will be rendered diagram and a window containing the calculated results.

Key words

Three-phase system, phasor, phasor diagram, phase, current, voltage, power supply, load, conduction, impedance, admittance, resistance, inductance, capacitance, frequency, star, delta, Java, Swing.

Seznam použitých zkratk

Symbol	Název	Jednotka
I	elektrický proud	(A)
I_A, I_B, I_C, I_0	elektrický proud fází	(A)
i_1, i_2, i_3	elektrický proud	(A)
I_s	ustřední napětí	(A)
I_f	fázové napětí	(A)
I_X	proud určené fáze	(A)
U	elektrické napětí	(V)
u_A, u_B, u_C	okamžité napětí	(V)
U_{mA}, U_{mB}, U_{mC}	indukovaná napětí na fázi	(V)
U_A, U_B, U_C, U_U	napětí zdroje	(V)
U_X	napětí určené fáze	(V)
U_{NA}, U_{NB}, U_{NC}	napětí fází a nulovým vodičem	(V)
U_{AB}, U_{BC}, U_{CA}	napětí mezi danými fázemi	(V)
U_{XY}	napětí určené mezi danými fázemi	(V)
U_s	ustřední napětí	(V)
U_f	fázové napětí	(V)
U_0	napětí v bodě nula	(V)
U_R	rozdíl napětí zdroje a bodu nula	(V)
ω	úhlový kmitočet	
Z_X	impedance pro určitou fázi	(Ω)
Z_U	impedance zdroje	(Ω)
Z_V	impedance vedení	(Ω)
Z_Z	impedance zátěže	(Ω)
Z_0	impedance nulového vodiče	(Ω)
Z_A, Z_B, Z_C	impedance určené fáze	(Ω)

Z_{AB}, Z_{BC}, Z_{AC}	impedance dvou fází	(Ω)
<u>Symbol</u>	<u>Název</u>	<u>Jednotka</u>
Z_{ZA}, Z_{ZB}, Z_{ZC}	impedance zdroje určené fáze	(Ω)
Z_{VA}, Z_{VB}, Z_{VC}	impedance vedení určené fáze	(Ω)
Z_1, Z_2, Z_3	impedance zátěže určené fáze	(Ω)
Y	admitance	(S)
Y_X	admitance pro určitou fázi	(S)
$Y_0,$	admitance nulového vodiče	(S)
Y_A, Y_B, Y_C	admitance určené fáze	(S)
T	perioda	(s)
A, B, C	začátek vinutí fáze	
A', B', C'	konec vinutí fáze	
Y	hvězda	
D	trojúhelník	
YY	hvězda – hvězda	
YY3	hvězda – hvězda 3 vodiče	
YY4	hvězda – hvězda 4 vodiče	
YD	hvězda – trojúhelník	
DY	trojúhelník – hvězda	
DD	trojúhelník	– trojúhelník

Obsah

1 Úvod.....	3
2 Přenosové sítě	4
2.1 Přenosové sítě z hlediska počtu fází.....	4
2.2 Trojfázová soustava	4
2.3 Trojfázové zdroje	4
2.4 Trojfázové zátěže	5
2.5 Způsoby zapojení trojfázových soustav	6
2.5.1 Zdroje	6
2.5.1.1 Zapojení zdrojů do hvězdy	6
2.5.1.2 Zapojení zdrojů do trojúhelníku	8
2.5.2 Zátěže	9
2.5.2.1 Zapojení zátěží do hvězdy	9
2.5.2.2 Zapojení zátěží do trojúhelníku	10
2.5.3 Vedení	11
2.6 Typy zapojení.....	11
2.6.1 Zapojení soustavy do hvězdy – hvězdy (YY)	11
2.6.2 Zapojení soustavy do hvězdy – hvězdy se třemi vodiči (YY3)	12
2.6.3 Zapojení soustavy do hvězdy – hvězdy se čtyřmi vodiči (YY4)	14
2.6.4 Zapojení soustavy do hvězdy – trojúhelníku (YD)	15
2.6.5 Zapojení soustavy do trojúhelníku – trojúhelníku (DD)	16
2.6.6 Zapojení soustavy do trojúhelníku – hvězda (DY)	17
3 Základní popis funkce programu	19
3.1 Grafické rozhraní programu	19
3.2 Popis diagramu	21
3.3 Ovládání programu	22
4 Implementace programu.....	23

4.1 Programovací jazyk Java.....	24
4.2 Aplikační doména	24
4.2.1 Třída KomplexniCislo.....	24
4.2.2 Třída Faze	24
4.3 Aplikační logika.....	24
4.3.1 Třída KomplexniCisloKalkulator.....	25
4.3.2 Třída ZapojeniKalkulator.....	26
4.4 Grafické rozhraní	34
4.4.1 Implementace vykreslení diagramu	34
4.4.2 Implementace zobrazení výsledků	35
4.4.3 Interakce s uživatelem.....	35
5 Závěr	42
6 Literatura.....	43
7 Přílohy.....	44

1 Úvod

Cílem této práce je vytvoření programu, který bude umět spočítat a následně vykreslit fázový diagram napětí a proudů trojfázové soustavy na počítači. Vytvořený program by měl pracovat s různými typy zapojení trojfázových zdrojů a zátěží, dále by měl počítat i s impedancemi na zdroji, vedení a zátěži. Impedanci bude možné zadat buď přímo hodnotou pokud ji známe, nebo zadáním parametrů jednotlivých zátěží, které budou představovat odpor, indukčnost, kapacita a frekvence zdrojů. U zdroje se bude moci dále nastavovat napětí.

První teoretická část práce popisuje přenosové sítě se zaměřením na nejpoužívanější trojfázovou soustavu, trojfázové zdroje, zátěže, vedení, typy zapojení.

V druhé části je zachycen návrh grafického rozhraní a jeho fungování.

V další části je popsána funkce programu a jeho implementace.

2 Přenosové sítě

2.1 Přenosové sítě z hlediska počtu fází

Přenosové sítě mohou být tvořeny různým počtem fází, avšak nejpoužívanější je trojfázová soustava. Dvoufázová soustava má oproti třífázové soustavě nevýhodu v určité možnosti, že se otáčející rotor spotřebiče zastaví na mrtvém bodě, toto u trojfázové soustavy nehrozí. Čtyřfázové a vícefázové soustavy mají oproti trojfázové soustavě výhodu větší plynulosti, avšak je u nich potřeba více materiálů.

Hlavní výhodou trojfázové soustavy je především její hospodárnější přenos. Toho je dosaženo použitím menšího počtu vodičů, kdy se jednotlivé fáze spojí do hvězdy nebo do trojúhelníku a tím místo šesti vodičů stačí tři, nebo čtyři vodiče. U trojfázových soustav je rovněž vyšší efektivita přenosu.

Další výhodou trojfázové soustavy je snadný způsob vytvoření točivého magnetického pole a jeho využití zejména v asynchronních motorech [1]. Výhodou je také to, že se při zapojení trojfázové soustavy vyskytují dva druhy napětí, a to fázové a sdružené. [1,2,3]

2.2 Trojfázová soustava

Trojfázová soustava vznikne jednoduchým spojením trojfázového zdroje a trojfázové zátěže. Existuje několik druhů zapojení této soustavy. Tyto typy jsou hvězda – hvězda (YY), hvězda – trojúhelník (YD), trojúhelník – hvězda (DY), trojúhelník – trojúhelník (DD). Vedení se skládá většinou ze tří vodičů. V zapojení typu hvězda – hvězda (YY) existuje možnost dvou zapojení, a to se třemi vodiči (YY3) anebo pomocí přidání čtvrtého, nulového vodiče (YY4). V současnosti je čtvrtý nulový vodič rozdělen na vodiče dva, a to nulový vodič a zemnicí vodič. Tedy při zapojení typu hvězda – hvězda se často používá pět vodičů. Trojfázový proud se generuje především pomocí alternátoru, jenž je tvořen pomocí tří cívek, které jsou navzájem otočeny vůči sobě o 120° . Trojfázová soustava se používá především v motorech. Další využití této soustavy je u transformátoru. Zde je jeho hlavní výhodou efektivní přenos elektrické energie. [1,2,3]

2.3 Trojfázové zdroje

Trojfázový zdroj se skládá ze tří jednofázových zdrojů. Jednotlivé fáze generují určitou amplitudu napětí o určité frekvenci, a toto napětí je vůči sobě posunuté o úhel ϕ . Tyto parametry se nemusí shodovat, ale je výhodnější, pokud mají stejné napětí o stejné frekvenci, potom se tomuto zdroji říká zdroj souměrný. V alternátoru, jenž má cívky natočené o 120° tak většinou vzniká napětí, které je vůči sobě také posunuto o 120° . Frekvence závisí na otáčkách elektromagnetu rotoru, jelikož jedna otočka motoru se rovná jedné periodě indukovaného napětí.

Rotor je tvořený v podstatě elektromagnetem, napájeným jednosměrným proudem a je otáčený okolo osy. Stator z dynamických plechů má po obvodu šest drážek, obvykle s roztečí 60° , v kterých jsou umístěna vinutí jednotlivých fází 1 – 1', 2 – 2', 3 – 3', kde 1, 2, 3 jsou začátky a 1', 2', 3' jsou konce fází. Sled fází 1, 2, 3 je takový, že při otáčení rotoru se dostává jeho severní pól nejprve na začátek fáze 1, potom 2 a nakonec 3. V jednotlivých vinutích statoru se indukují následkem pohybu elektromagnetické indukce střídavé napětí o okamžitých hodnotách u_1 , u_2 , u_3 , které mají všechny stejnou frekvenci.[1]

Fázový posun vzniká tak, že se rotor dostane na začátek další fáze se zpožděním $\frac{T}{3}$ (T je doba jedné otočky rotoru). Taková soustava se dá vyjádřit pomocí následujících vzorců: [1]

$$u_A = U_{mA} \cdot \sin(\omega t) \quad (1)$$

$$u_B = U_{mB} \cdot \sin(\omega t - 120^\circ)$$

$$u_C = U_{mC} \cdot \sin(\omega t + 120^\circ)$$

kde: u_A , u_B , u_C – okamžitá hodnota,

U_{mA} , U_{mB} , U_{mC} – napětí indukovaná napětí na jednotlivých fázích,

$\sin(\omega t)$ – fázový posun o daný úhel.

Pokud je součet okamžitých hodnot jednotlivých fázových napětí v jednom okamžiku roven nule, tak je soustava souměrná. [1]

$$u_A + u_B + u_C = 0 \quad (2)$$

kde: u_A , u_B , u_C – okamžitá hodnota. [1,2,3]

2.4 Trojfázové zátěže

Trojfázová zátěž se skládá z tří složek, dohromady tyto složky vytvářejí impedanci. Impedance se skládá z reálné části, kterou určuje odpor zátěže, a dále z imaginární části, kterou určuje indukčnost a kapacita zátěže. Indukčnost zvyšuje imaginární hodnotu a kapacita ji naopak zmenšuje. Pokud mají impedance stejně velkou komplexní hodnotu, pak jsou zátěže souměrné. Jednotlivé složky zátěže se mohou od sebe lišit, důležité je, aby byly rovny komplexní hodnoty impedance. Pokud se k souměrné zátěži připojí souměrný zdroj, vznikne souměrná soustava, kterou protéká souměrný proud, nezávisle na způsobu zapojení zdroje a zátěže.

Trojfázová zátěž je často tvořena jedním konstrukčním celkem. Příkladem je v technické praxi jednoduchý a ve střídavých pohonech často používaný asynchronní motor, jehož princip činnosti je opačný k funkci alternátoru. Trojfázová napájecí soustava připojená k vinutím A – A', B – B', C – C' s proudy i_1 , i_2 a i_3 vytváří točivé magnetické pole, jehož výslednici si můžeme představit jako magnet, otáčející se proti směru oběhu sousledné nebo zpětné napájecí soustavy úhlovou rychlostí (kmotočtem) ω této napájecí soustavy[2]. [1,2,3]

2.5 Způsoby zapojení trojfázových soustav

2.5.1 Zdroje

Zdroje se zapojují do hvězdy, nebo do trojúhelníku, dále existuje zapojení do lomené hvězdy. Jsou-li zdroje zapojeny souměrně, tak získáme hodnoty napětí vycházející ze vztahu (2). Reálné zdroje mají na rozdíl od ideálních zdrojů svou vlastní impedanci, která ovlivňuje soustavu. Další věcí, která ovlivňuje impedanci, je fázový posun napětí, kdy indukčnost napětí impedanci zvyšuje a kapacita ji naopak zmenšuje. [1,2]

2.5.1.1 Zapojení zdrojů do hvězdy

Princip zapojení zdrojů zapojených do hvězdy spočívá v tom, že se spojí konce nebo začátky všech zdrojů dohromady. U zdrojů do hvězdy je možné vyvedení tří nebo čtyř vodičů. Soustava se čtyřmi vodiči má výhodu ve využití dvou různých napětí s různými efektivními hodnotami. Tato napětí jsou mezi jednotlivými fázemi nebo mezi fází a nulovým vodičem N. Podle 2. Kirchhoffova zákona platí: [1]

$$U_{NA} - U_A = 0 \quad (3)$$

$$U_{NB} - U_B = 0$$

$$U_{NC} - U_C = 0$$

kde: U_A, U_B, U_C – napětí zdroje,

U_{NA}, U_{NB}, U_{NC} – napětí mezi nulovým vodičem a danou fází.

Soustava, jež neobsahuje čtvrtý nulový vodič, má pouze soustavu napětí, která jsou mezi jednotlivými fázemi. Pro tuto soustavu platí: [1]

$$U_{AB} - U_A + U_B = 0 \quad (4)$$

$$U_{BC} - U_B + U_C = 0$$

$$U_{CA} - U_C + U_A = 0$$

kde: U_A, U_B, U_C – napětí zdroje,

U_{AB}, U_{BC}, U_{CA} – napětí mezi danými fázemi.

Soustava napětí mezi fází a nulovým bodem N se nazývá fázové napětí a soustava napětí, jež se nachází mezi dvěma fázemi, se nazývá sdružené napětí. Sdružené napětí je $\sqrt{3}$ krát větší než fázové a má také vůči fázovému napětí posunutou fází o 30° .

$$U_s = U_f \cdot \sqrt{3} \quad (5)$$

kde: U_s – sdružené napětí, [1]

U_f – fázové napětí.

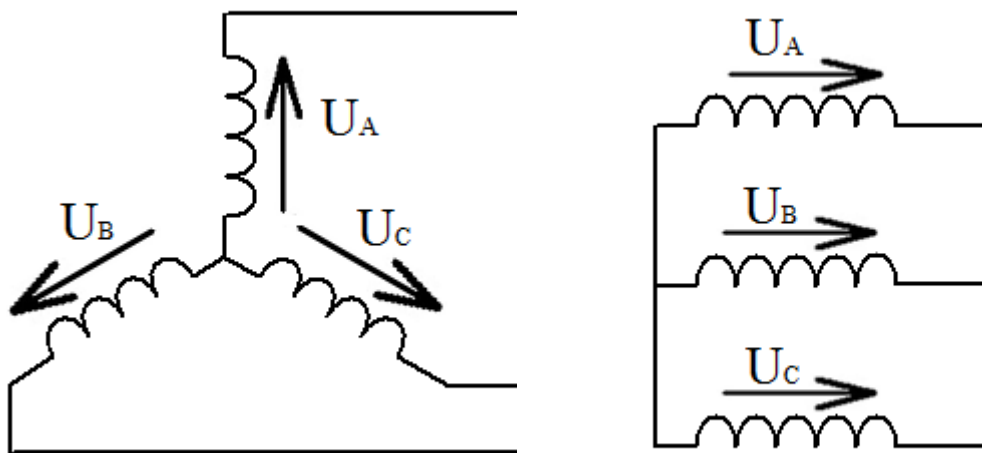
Proud protékající čtvrtým nulovým vodičem je podle Kirchhoffova zákona: [1]

$$I_0 = I_A + I_B + I_C \quad (6)$$

kde: I_0 – proud procházející čtvrtým nulovým vodičem,

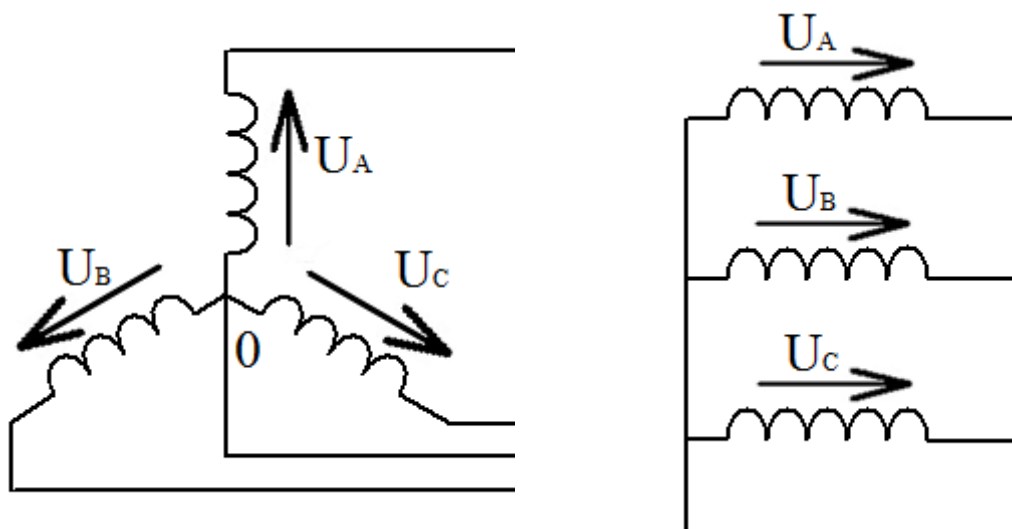
I_A, I_B, I_C – proudy procházející jednotlivými fázemi.

Pokud jsou proudy jednotlivých fází stejně velké, proud protékající čtvrtým nulovým vodičem je potom roven nule. V tom případě se fázový proud rovná proudu sdruženému.



Obr. 1.a Technické znázornění zapojení zdroje do hvězdy se třemi vodiči

Obr. 1.b Schématické znázornění zapojení zdroje do hvězdy se třemi vodiči



Obr. 2.a Technické znázornění zapojení zdroje do hvězdy se čtyřmi vodiči

Obr. 2.b Schématické znázornění zapojení zdroje do hvězdy se čtyřmi vodiči

[1,2]

2.5.1.2 Zapojení zdrojů do trojúhelníku

Princip zapojení zdrojů zapojených do trojúhelníku spočívá v tom, že začátek zdroje je spojen s koncem dalšího zdroje a je vyveden vývod na vedení. V této soustavě je jeden druh napětí, a tím je napětí sdružené. Podle 2. Kirchhoffova zákona platí: [1]

$$U_{AB} - U_A = 0 \quad (7)$$

$$U_{BC} - U_B = 0$$

$$U_{CA} - U_C = 0$$

kde: U_A, U_B, U_C – napětí zdroje,

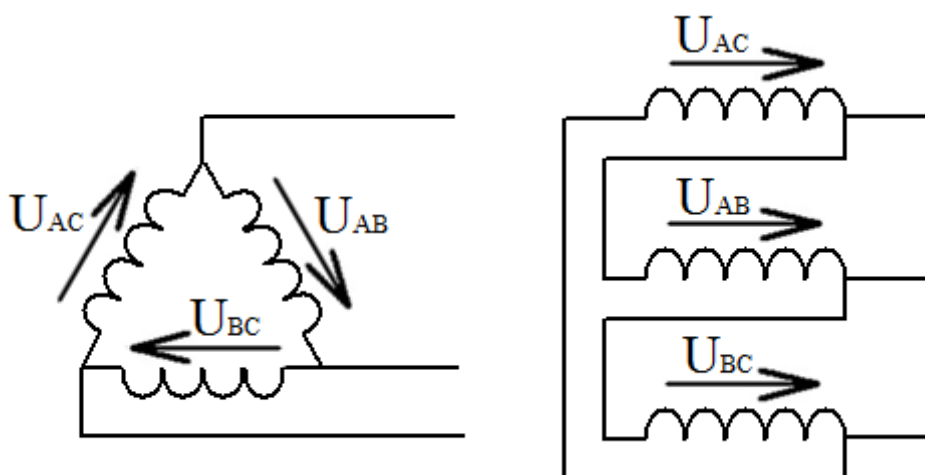
U_{AB}, U_{BC}, U_{CA} – napětí mezi danými fázemi.

Z tohoto vztahu vyplývá, že napětí ze zdroje se rovnají svým sdruženým napětím. Podobně jako u zapojení zdroje do hvězdy, kde je vztah mezi fázovým a sdruženým napětím, je zde vztah mezi fázovým a sdruženým proudem: [1]

$$I_s = I_f \cdot \sqrt{3} \quad (8)$$

kde: I_s – sdružený proud,

I_f – fázový proud.



Obr. 3.a Technické znázornění zapojení zdroje do trojúhelníku

Obr. 3.b Schématické znázornění zapojení zdroje do trojúhelníku

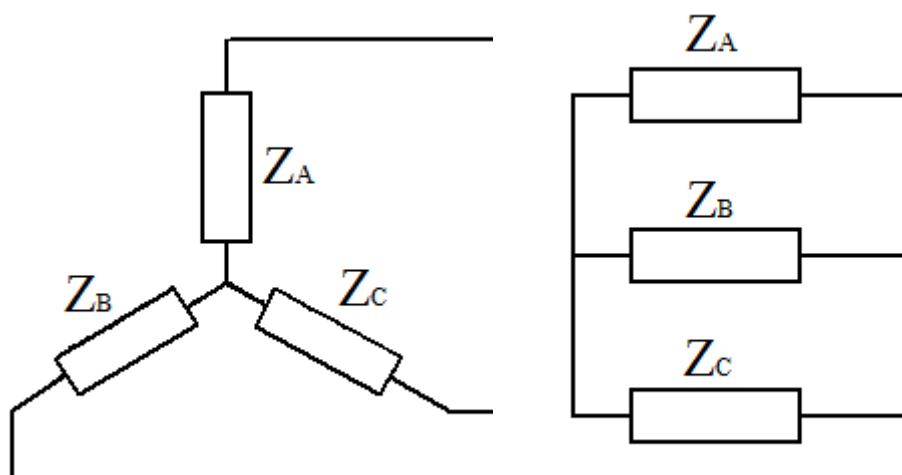
[1,2]

2.5.2 Zátěže

U zátěží jsou stejná zapojení, a to do hvězdy nebo do trojúhelníku. Zapojení zátěží však lze transfigurovat z jednoho typu zapojení na druhý. [1,2]

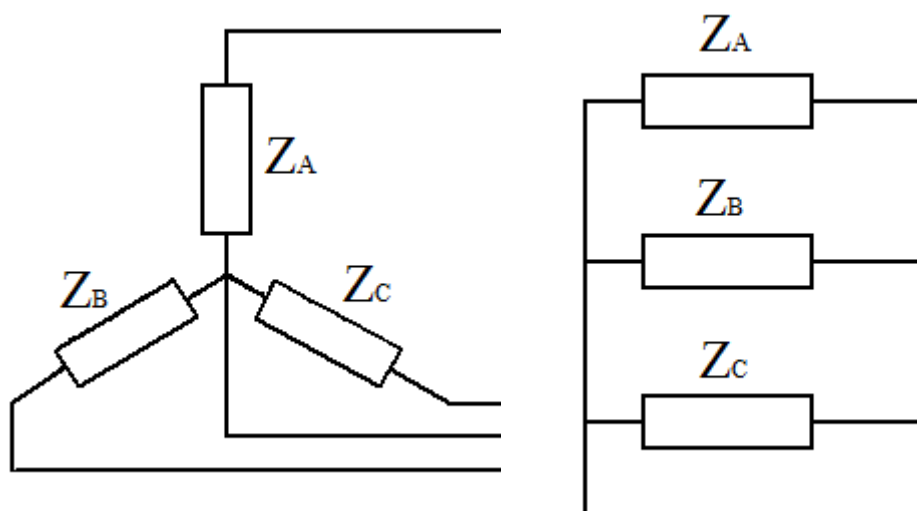
2.5.2.1 Zapojení zátěží do hvězdy

Stejně jako u zdrojů zapojení zátěží do hvězdy vznikne spojením začátků nebo konců zátěže. Tyto zátěže bývají většinou pasivní a spojený bod označujeme 0. Tato soustava se může připojit ke zdroji pomocí tří nebo čtyř vodičů s tím, že čtvrtý nulový vodič je připojen k společnému bodu 0.



Obr. 4.a Technické znázornění zapojení zátěže do hvězdy se třemi vodiči

Obr. 4.b Schématické znázornění zapojení zátěže do hvězdy se třemi vodiči



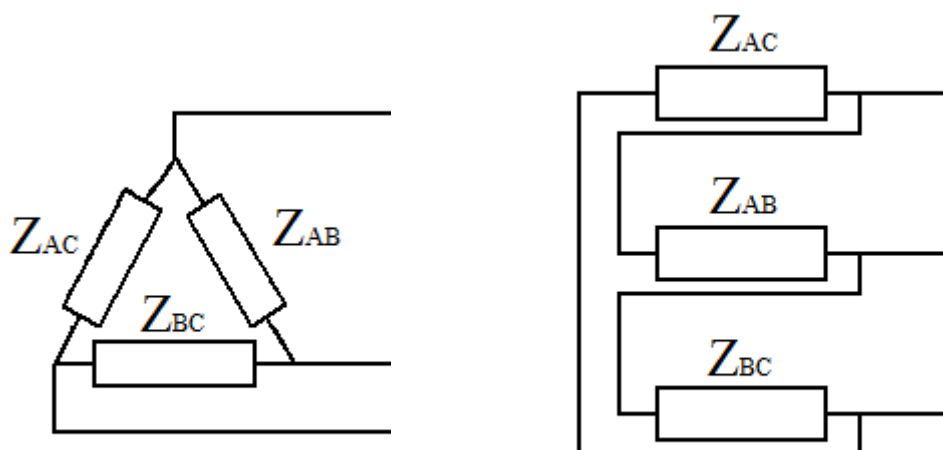
Obr. 5.a Technické znázornění zapojení zátěže do hvězdy se čtyřmi vodiči

Obr. 5.b Schématické znázornění zapojení zátěže do hvězdy se čtyřmi vodiči

[1,2]

2.4.2.2 Zapojení zátěží do trojúhelníku

Stejně jako u zdrojů zapojení zátěží do trojúhelníku vznikne tak, že spojíme konec jedné zátěže se začátkem druhé. Tuto soustavu lze kde zdroji připojit pouze třemi vodiči.



Obr. 6.a Technické znázornění zapojení zátěže do trojúhelníku

Obr. 6.b Schématické znázornění zapojení zátěže do trojúhelníku

[1,2]

2.5.3 Vedení

Další věc, jež ovlivňuje soustavu, je vedení, které má svou vlastní impedanci, která se nachází mezi zdrojem a zátěží. Vedení se pro svou nízkou impedanci většinou nezapočítává, nebo se berou jenom některé konkrétní parametry. Počet linek tvořící vedení jsou tři nebo čtyři vodiče. [1,2]

2.6 Typy zapojení

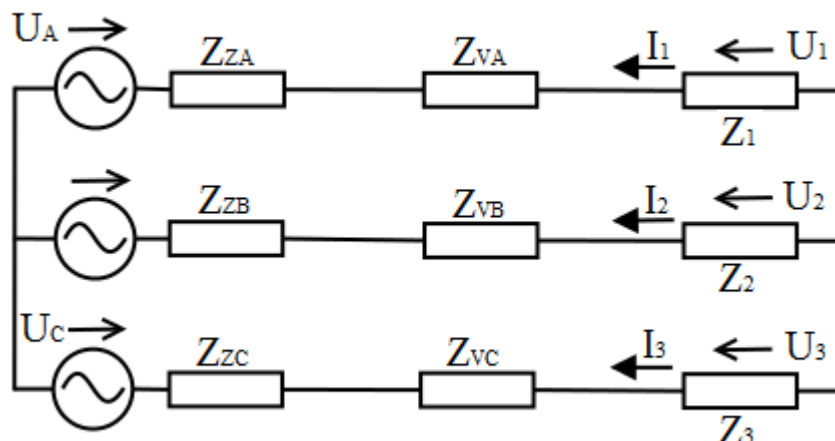
Typy zapojení soustavy jsou tvořeny kombinacemi výše uvedených zapojení zdrojů a zátěží. Jedním typem soustavy je soustava zapojena do hvězdy – hvězdy, tato soustava se ještě dělí na soustavu zapojenou se třemi vodiči a soustavu se čtyřmi vodiči, kde čtvrtý vodič je vodič nulový. Dalšími typy soustavy jsou soustavy zapojeny do hvězdy – trojúhelníku, trojúhelníku – hvězdy a trojúhelníku – trojúhelníku. Charakteristiky těchto soustav jsou určeny pomocí zapojení jednotlivých prvků, tak jak jsou popsány v předchozích kapitolách o zapojení zdrojů a zátěží. [1,2]

2.6.1 Zapojení soustavy do hvězdy – hvězdy (YY)

Tato soustava je jediná, u které je možné zapojení čtvrtého nulového vodiče. Velkou výhodou této soustavy je, že se jednotlivé fáze navzájem neovlivňují. Toto je možné díky tomu, že

v bodě nula se budou jednotlivá fázová a sdružená napětí rovnat nule. Ze zapojení zdroje a zátěže do hvězdy se zjistí, že v této soustavě se nacházejí dva typy napětí, a to je fázové a sdružené. [1,2]

2.6.2 Zapojení soustavy do hvězdy – hvězdy se třemi vodiči (YY3)



Obr. 7 Soustava zapojena do hvězdy – hvězdy se třemi vodiči

Jelikož se všechny soustavy řeší pomocí komplexních čísel, tak se nejprve musí převést napětí ze zdroje na jeho komplexní tvar pomocí vztahu: [4]

$$U_U = U_U \cdot \sin(\text{úhel}) \quad (9)$$

kde: U_U – napětí na zdroji,

úhel – úhel napětí, normálně se zadává 0° , 120° a 240° .

Poté zjistíme celkovou impedanci dané fáze: [4]

$$Z_X = Z_U + Z_V + Z_Z \quad (10)$$

kde: Z_X – celková impedance pro fázi x,

Z_U – impedance zdroje,

Z_V – impedance vedení,

Z_Z – impedance zátěže.

Takto lze spočítat impedanci celého vedení pouze v případě zapojení soustavy do hvězdy – hvězdy, a to z toho důvodu, že v této soustavě se jednotlivé fáze neovlivňují.

Dalším krokem je spočtení admitance: [4]

$$Y_X = \frac{1}{Z_X} \quad (11)$$

kde: Y_X – celková admitance pro fázi x,

Z_X – celková impedance pro fázi x.

Jak lze vidět ze vztahu, admitance je jenom převrácená hodnota impedance a slouží k popisu zdánlivé vodivosti součástek a jejich fázovému posunu napětí vůči proudu.

Nyní se zjistí, jaké je napětí v bodě nula: [4]

$$U_0 = \frac{U_A * Y_A + Y_B * U_B + Y_C * U_C}{Y_A + Y_B + Y_C} \quad (12)$$

kde: U_0 – napětí v bodě nula,

Y_1, Y_2, Y_3 – celková admitance dané fáze,

U_A, U_B, U_C , - napětí na zdrojích.

Dále se spočítá rozdíl mezi napětím na zdroji a v bodě nula: [4]

$$U_R = U_X - U_0 \quad (13)$$

kde: U_R – rozdíl mezi napětím na zdroji a v bodě nula,

U_X – napětí dané fáze,

U_0 – napětí v bodě nula.

Nyní se spočítá, jak velký proud prochází danou fází: [4]

$$I_X = Y_X \cdot U_R \quad (14)$$

kde: I_X – proud protékající danou fází,

Y_X – celková impedance pro danou fázi,

U_R – rozdíl mezi napětím na zdroji a v bodě nula.

A s pomocí tohoto proudu je možné vypočítat napětí nacházející se na jednotlivých částech fáze: [4]

$$U_{XY} = I_X \cdot Z_{XY} \quad (15)$$

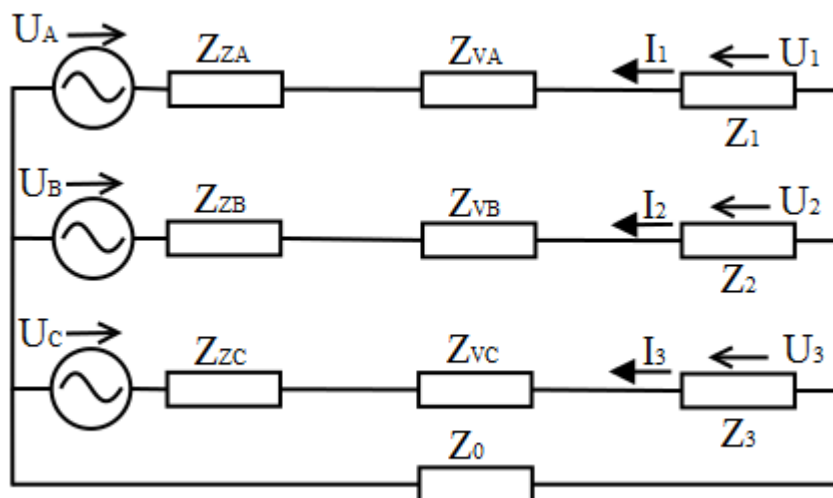
Kde: U_X – napětí na dané fázi nacházející se na dané části,

I_X – proud protékající danou fází,

Z_X – impedance fáze fáze a části.

V tomto vztahu se počítá napětí nacházející se na zdroji, vedení a zátěži. Místo, pro které se vypočítá napětí, se zvolí podle jeho impedance a proudu. Tedy pokud se bude počítat napětí nacházející se na vedení fáze B, tak se použije impedance vedení pro fázi B a proud fáze B. Takto se spočítá soustava zapojena do hvězdy – hvězdy se třemi vodiči. [1,2]

2.6.3 Zapojení soustavy do hvězdy – hvězdy se čtyřmi vodiči (YY4)



Obr. 8 Soustava zapojena do hvězdy – hvězdy se třemi vodiči

Tato soustava se počítá podobně jako soustava zapojena do hvězdy – hvězdy se třemi vodiči. Nejprve se převede napětí na komplexní tvar pomocí vztahu (9), poté se zjistí celková impedance jednotlivých fází pomocí vztahu (10) a následně se vypočte celková admitance jednotlivých fází pomocí vztahu (11). Zde dochází ke změně v tom, že se spočítá také admitance čtvrtého nulového vodiče pomocí vztahu: [4]

$$Y_0 = \frac{1}{Z_0} \quad (16)$$

kde: Y_0 – admitance pro čtvrtý nulový vodič,

Z_0 – impedance pro čtvrtý nulový vodič.

Rozdíl je v tom, že se nemusí počítat impedance zdroje a zátěže a to proto, že na tomto vedení není žádný zdroj ani žádná zátěž. Jediná impedance je vedení.

Nyní se spočítá, jaká je velikost napětí v bodě nula pomocí vztahu: [4]

$$U_0 = \frac{U_A * Y_A + Y_B * U_B + Y_C * U_C}{Y_A + Y_B + Y_C + Y_0} \quad (17)$$

kde: U_0 – napětí v bodě nula,

Y_1, Y_2, Y_3 – celková admitance dané fáze,

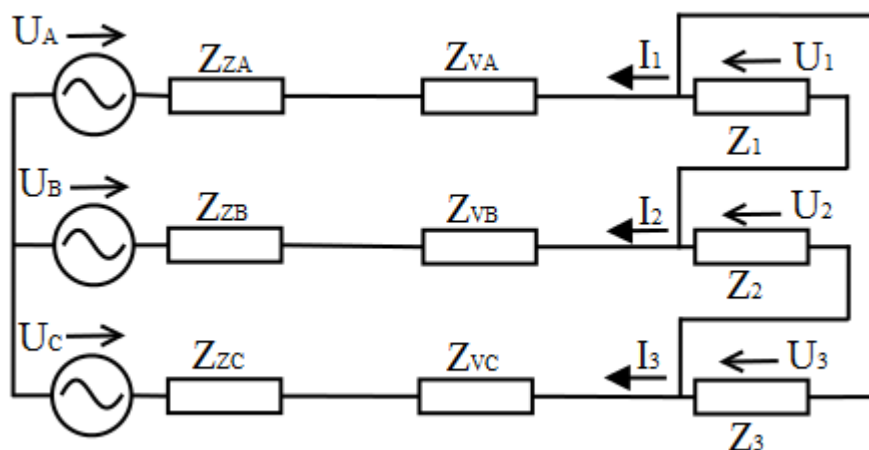
U_A, U_B, U_C , - napětí na zdrojích,

Y_0 – admitance čtvrtého nulového vodiče.

Čtvrtý nulový vodič slouží především k zabránění posunutí nuly v diagramu pro napětí na zátěži, jak lze vidět ze vztahu (16) čím menší je impedance, tím je větší admitance. A čím je

větší admitance, tím je menší napětí v bodě nula, čím je menší napětí v bodě nula, tím je menší posun nuly pro napětí na zátěži v diagramu. Dále se počítá stejně jako pro soustavu zapojenou do hvězdy – hvězdy se třemi vodiči a to tak, že pomocí vztahu (13) se spočítá rozdíl napětí mezi napětím zdroje a napětím v bodě nula. Poté se spočítají jednotlivé proudy pro fáze pomocí vztahu (14). A nakonec pomocí vztahu (15) se vypočítají jednotlivá napětí pro jednotlivé části soustavy. [1,2]

2.6.4 Zapojení soustavy do hvězdy – trojúhelníku (YD)



Obr. 9 Soustava zapojena do hvězdy – trojúhelníku

Tato soustava se řeší pomocí výpočtu pro zapojení soustavy do hvězdy – hvězdy se třemi vodiči. Nejprve se musí převést impedance zátěže ze zapojení do trojúhelníku do hodnot, které by měly při zapojení do hvězdy, toto se provede pomocí následujících vztahů: [5]

$$Z_A = \frac{Z_{AB} \cdot Z_{AC}}{Z_{AB} + Z_{AC} + Z_{BC}} \quad (18)$$

$$Z_B = \frac{Z_{AB} \cdot Z_{BC}}{Z_{AB} + Z_{AC} + Z_{BC}}$$

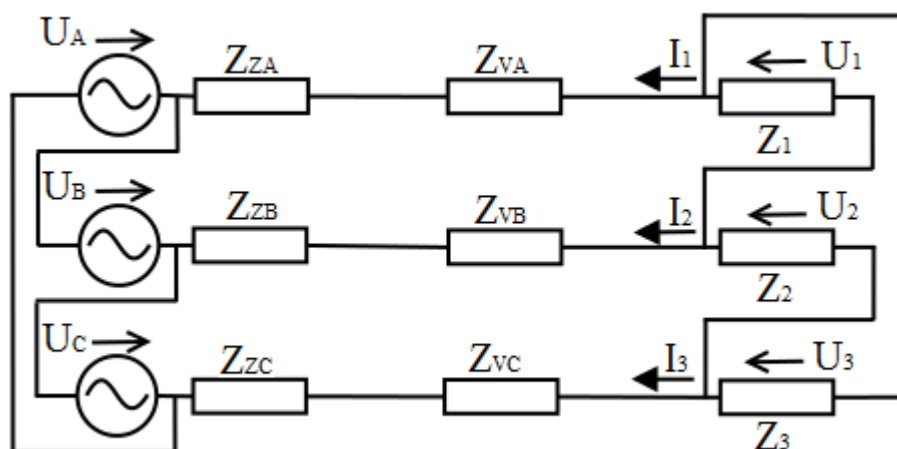
$$Z_C = \frac{Z_{AC} \cdot Z_{BC}}{Z_{AB} + Z_{AC} + Z_{BC}}$$

kde: Z_A, Z_B, Z_C – impedance pro dané fáze,

Z_{AB}, Z_{BC}, Z_{AC} – impedance pro dvě spojené fáze.

Tento převod z hodnot zátěží zapojených do trojúhelníku na hodnoty pro zapojení do hvězdy je zde proto, aby bylo možné spočítat celkovou impedanci dané fáze. Po tomto převodu se soustava dále řeší pomocí postupu pro zapojení soustavy do hvězdy – hvězdy se třemi vodiči (kapitola 2.4.4.2). [1,2]

2.6.5 Zapojení soustavy do trojúhelníku – trojúhelníku (DD)



Obr. 10 Soustava zapojena do trojúhelníku – trojúhelníku

Aby se tato soustava mohla řešit, bude se předpokládat, že platí následující podmínka:

$$I_A + I_B + I_C = 0 \quad (19)$$

kde: I_A, I_B, I_C – proudy jednotlivých fází.

Tato soustava se řeší pomocí pěti rovnic o pěti neznámých:

$$\begin{aligned} & \left(\frac{1}{Z_A} + \frac{1}{Z_C} + \frac{1}{Z_{V1}} \right) \cdot U_{V1} - \frac{U_{V2}}{Z_A} - \frac{U_{V3}}{Z_C} + \left(\frac{1}{Z_A} + \frac{1}{Z_C} \right) \cdot U_1 + \frac{U_2}{Z_C} = \frac{U_A}{Z_A} - \frac{U_C}{Z_C} \quad (20) \\ & - \frac{U_{V1}}{Z_A} + \left(\frac{1}{Z_A} + \frac{1}{Z_B} + \frac{1}{Z_{V2}} \right) \cdot U_{V2} - \frac{U_{V3}}{Z_B} - \frac{U_1}{Z_A} + \frac{U_2}{Z_B} = \frac{U_B}{Z_B} - \frac{U_A}{Z_A} \\ & - \frac{U_{V1}}{Z_C} - \frac{U_{V2}}{Z_B} + \left(\frac{1}{Z_B} + \frac{1}{Z_C} + \frac{1}{Z_{V3}} \right) \cdot U_{V3} - \frac{U_1}{Z_C} - \left(\frac{1}{Z_B} + \frac{1}{Z_C} \right) \cdot U_2 = \frac{U_C}{Z_C} - \frac{U_B}{Z_B} \\ & \left(\frac{1}{Z_A} + \frac{1}{Z_C} \right) \cdot U_{V1} - \frac{U_{V2}}{Z_A} - \frac{U_{V3}}{Z_C} + \left(\frac{1}{Z_1} + \frac{1}{Z_3} + \frac{1}{Z_A} + \frac{1}{Z_C} \right) \cdot U_1 + \left(\frac{1}{Z_3} + \frac{1}{Z_C} \right) \cdot U_2 = \\ & \quad \frac{U_A}{Z_A} - \frac{U_C}{Z_C} \\ & \frac{U_{V1}}{Z_C} + \frac{U_{V2}}{Z_B} - \left(\frac{1}{Z_B} + \frac{1}{Z_C} \right) \cdot U_{V3} + \left(\frac{1}{Z_3} + \frac{1}{Z_C} \right) \cdot U_1 + \left(\frac{1}{Z_2} + \frac{1}{Z_3} + \frac{1}{Z_B} + \frac{1}{Z_C} \right) \cdot U_2 = \frac{U_B}{Z_B} - \\ & \quad \frac{U_C}{Z_C} \end{aligned}$$

Pro jednodušší řešení se budou tyto rovnice řešit pomocí matic a jejich postupným rozkladem na jednotkovou matici. Těchto pět rovnic se dosadí do matice:

$$\begin{bmatrix} \frac{1}{\hat{Z}_A} + \frac{1}{\hat{Z}_C} + \frac{1}{\hat{Z}_{v1}} & -\frac{1}{\hat{Z}_A} & -\frac{1}{\hat{Z}_C} & \frac{1}{\hat{Z}_A} + \frac{1}{\hat{Z}_C} & \frac{1}{\hat{Z}_C} \\ -\frac{1}{\hat{Z}_A} & \frac{1}{\hat{Z}_A} + \frac{1}{\hat{Z}_B} + \frac{1}{\hat{Z}_{v2}} & -\frac{1}{\hat{Z}_B} & -\frac{1}{\hat{Z}_A} & \frac{1}{\hat{Z}_B} \\ -\frac{1}{\hat{Z}_C} & -\frac{1}{\hat{Z}_B} & \frac{1}{\hat{Z}_B} + \frac{1}{\hat{Z}_C} + \frac{1}{\hat{Z}_{v3}} & -\frac{1}{\hat{Z}_C} & -\frac{1}{\hat{Z}_B} - \frac{1}{\hat{Z}_C} \\ \frac{1}{\hat{Z}_A} + \frac{1}{\hat{Z}_C} & -\frac{1}{\hat{Z}_A} & -\frac{1}{\hat{Z}_C} & \frac{1}{\hat{Z}_1} + \frac{1}{\hat{Z}_3} + \frac{1}{\hat{Z}_A} + \frac{1}{\hat{Z}_C} & \frac{1}{\hat{Z}_3} + \frac{1}{\hat{Z}_C} \\ \frac{1}{\hat{Z}_C} & \frac{1}{\hat{Z}_B} & -\frac{1}{\hat{Z}_B} - \frac{1}{\hat{Z}_C} & \frac{1}{\hat{Z}_3} + \frac{1}{\hat{Z}_C} & \frac{1}{\hat{Z}_2} + \frac{1}{\hat{Z}_3} + \frac{1}{\hat{Z}_B} + \frac{1}{\hat{Z}_C} \end{bmatrix} \cdot \begin{bmatrix} \hat{U}_{v1} \\ \hat{U}_{v2} \\ \hat{U}_{v3} \\ \hat{U}_1 \\ \hat{U}_2 \end{bmatrix} = \begin{bmatrix} \frac{\hat{U}_A}{\hat{Z}_A} - \frac{\hat{U}_C}{\hat{Z}_C} \\ \frac{\hat{U}_B}{\hat{Z}_B} - \frac{\hat{U}_A}{\hat{Z}_A} \\ \frac{\hat{U}_C}{\hat{Z}_C} - \frac{\hat{U}_B}{\hat{Z}_B} \\ \frac{\hat{U}_A}{\hat{Z}_A} - \frac{\hat{U}_C}{\hat{Z}_C} \\ \frac{\hat{U}_B}{\hat{Z}_B} - \frac{\hat{U}_C}{\hat{Z}_C} \end{bmatrix}$$

Výsledkem bude napětí na všech vedeních a napětí na zátěži fáze 1 a 2. Napětí na poslední fázi se spočítá pomocí vztahu:

$$U_3 = -U_1 - U_2 \quad (21)$$

kde: U_1, U_2, U_3 – napětí jednotlivých fází na zátěži.

A z těchto hodnot je možné vypočítat napětí nacházející se na výstupu ze zdroje pomocí vztahů:

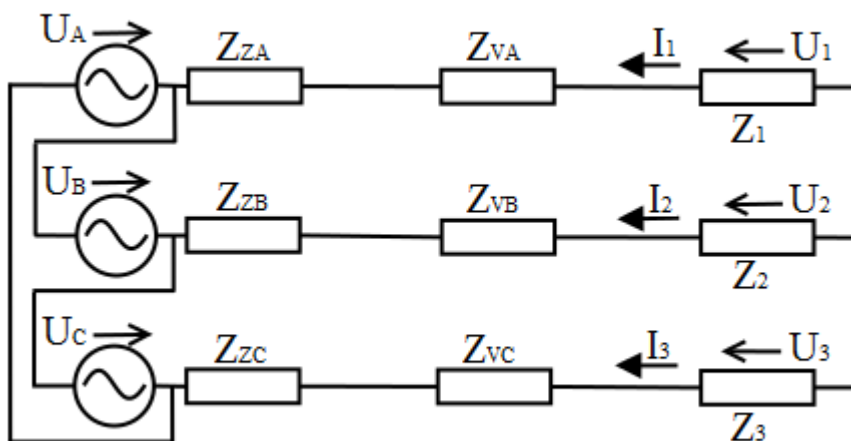
$$U_{ZA} = U_A - (U_{V1} + U_1 - U_{V2}) \quad (22)$$

$$U_{ZB} = U_B - (U_{V2} + U_2 - U_{V3})$$

$$U_{ZC} = U_C - (U_{V3} - U_{V1} - U_1 - U_2)$$

Takto se vypočítá soustava zapojená do trojúhelníku – trojúhelníku. [1,2]

2.6.6 Zapojení soustavy do trojúhelníku – hvězda (DY)



Obr. 11 Soustava zapojena do trojúhelníku – hvězdy

Tato soustava se řeší pomocí výpočtu pro zapojení soustavy do trojúhelníku – trojúhelníku. Nejprve se musí převést zátěže ze zapojení do hvězdy do hodnot, které by měly při zapojení do trojúhelníku, toto se provede pomocí následujících vztahů:

$$Z_{AB} = \frac{Z_A \cdot Z_B + Z_A \cdot Z_C + Z_B \cdot Z_C}{Z_C} \quad (23)$$

$$Z_{BC} = \frac{Z_A \cdot Z_B + Z_A \cdot Z_C + Z_B \cdot Z_C}{Z_A}$$

$$Z_{AC} = \frac{Z_A \cdot Z_B + Z_A \cdot Z_C + Z_B \cdot Z_C}{Z_B}$$

Po tomto převodu se soustava řeší stejně jako soustava zapojená do trojúhelníku – trojúhelníku. [1,2]

3 Základní popis funkce programu

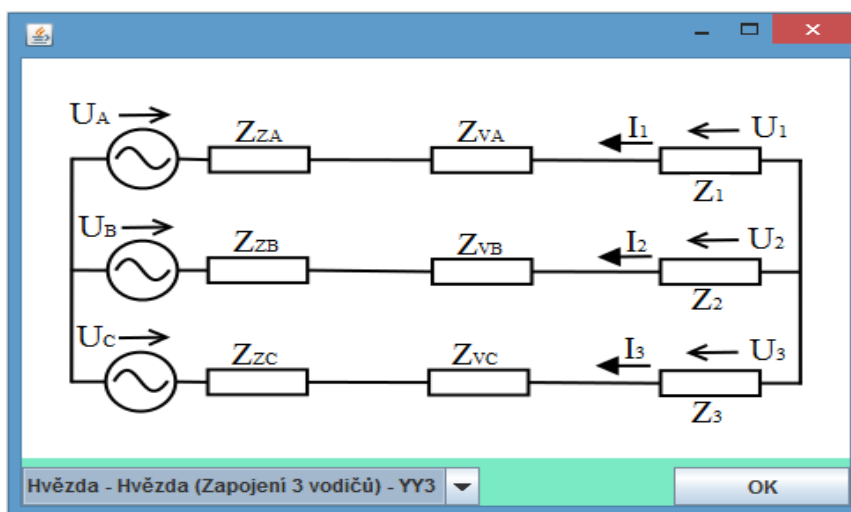
Výsledným cílem programu bude vytvoření fázorového diagramu, který bude obsahovat napětí zdroje, napětí na výstupu zdroje, napětí na vedení, napětí na zátěži, napětí v bodě nula a proudy jednotlivých fází. Dalším výstupem programu bude nové okno, ve kterém budou vypsány číselné hodnoty vypočtených napětí a proudů, a to ve formě algebraické, tak i ve formě goniometrické. Dále v programu bude možné nastavit fázový posun o určitý úhel (pro lepší přehlednost diagramu).

3.1 Grafické rozhraní programu

Okno programu je rozděleno na dvě části. Na levou stranu (obr. 12), kde jsou umístěna nastavení hodnot, typ zapojení, nastavení zobrazení jednotlivých fázorů a legenda popisující graf, a na pravou stranu okna (obr. 15), která slouží pro zobrazení diagramu. Toto rozdělení jsem zvolil z důvodu, že graf se vypisuje do kružnice, a proto stačí polovina okna pro nastavení hodnot a druhá pro zobrazení diagramu. Nastavení je zde proto, aby bylo možné rychle změnit hodnoty diagramu, ať z důvodu chybného zadání, nebo proto, že nevyhovují vypočtené hodnoty a vzhled diagramu.

Obr. 12 Levá strana hlavního okna programu (nastavení)

V nastavení hodnot se po stlačení tlačítka pro typ zapojení objeví okno (obr. 13), kde je možné vybrat typ zapojení soustavy. Výběr se provede pomocí volby konkrétního zadání z rozbalovacího seznamu. Nakonec se pomocí tlačítka OK okno zavře.



Obr. 13 Okno výběru typu zapojení

Dále v nastavení hodnot je možnost zadání impedance pomocí zadání jednotlivých prvků zátěže (platí také pro vedení a zdroj), pokud se zvolí, že se bude impedance takto zadávat. Po stisknutí tlačítka impedance zátěže (vedení nebo zdroje) se otevře okno (obr. 14), ve kterém se zadají hodnoty pro jednotlivé prvky pro všechny fáze. V prvním sloupci se zadávají hodnoty pro odpor zátěže, v druhém se zadává indukčnost a ve třetím sloupci se zadává kapacita. Pod jednotlivými sloupci se nachází rozbalovací seznamy, ve kterých se zvolí, pro jaký řád jsou hodnoty zadány. Jeden rozbalovací seznam určuje řád pro všechny hodnoty zadané v daném sloupci. Pomocí tlačítka OK se vypočítá impedance. Pro zadání impedance přímo se stiskne tlačítko Zrušit a okno se zavře bez provedení výpočtu impedance.

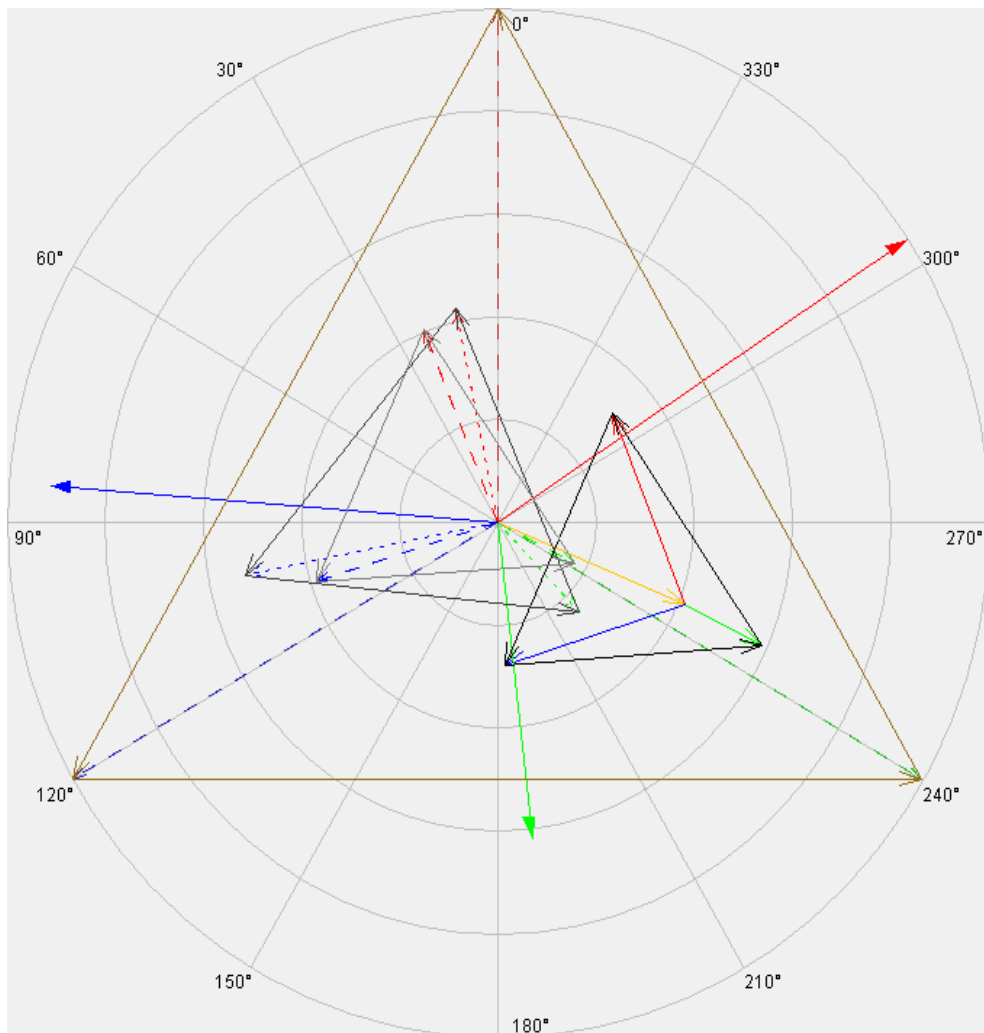
The 'Impedance' window contains three columns of input fields. The first column is for resistance (Ra, Rb, Rc) with values 12, 12, and 16. The second column is for inductance (La, Lb, Lc) with values 0, 55, and 0. The third column is for capacitance (Ca, Cb, Cc) with values 198, 999999, and 265. Below each column is a dropdown menu with a '-' symbol. At the bottom right are 'OK' and 'Zrušit' buttons.

Obr. 14 Okno pro zadání jednotlivých prvků impedance

Poslední okno se zobrazí poté, co se stiskne tlačítko Vypočíst a jsou zadány všechny hodnoty správně. V tomto okně se zobrazí výsledky. Toto okno je zde především proto, aby v hlavním okně programu nebylo příliš mnoho prvků, kvůli kterým by se musel zmenšit diagram, jelikož diagram je hlavním výstupem tohoto programu.

3.2 Popis diagramu

Na diagramu (obr. 15) lze vidět, že je tvořen kružnicemi, které určují velikost napětí. Těchto kružnic je zde pět. Každá kružnice představuje procenta napětí, tedy 20%, 40%, 60%, 80% a 100% napětí, kdy 100% napětí je rovno nejvyššímu napětí, které je možné najít ve výsledcích. Dále na diagramu lze vidět úsečky s počátkem ve středu souřadného systému, jež představují úhly fázového posunu, kdy každá úsečka označuje 30° po celé kružnici. Zde se nachází nula nahoře, nikoliv vpravo. Na diagramu je přerušovanou čarou zobrazováno napětí zdroje, kde tmavší čáry jsou zadaná napětí zdroje a světlejší jsou napětí, jež se nachází na výstupu ze zdroje. Dále přerušovaná čára představuje napětí, jež se nachází na vedení. A nakonec plná čára, která představuje napětí na zátěži. Plná čára, která má vyplněnou šipku, představuje proud. Oranžová ukazuje napětí nacházející se v bodě nula. Od šedé až po černou jsou čáry představující sdružené napětí nacházející se na zdroji (hnědá), výstupu zdroje (šedá), vedení (tmavě šedá) a zátěže (černá).



Obr. 15 Diagram třífázové sítě

3.3 Ovládání programu

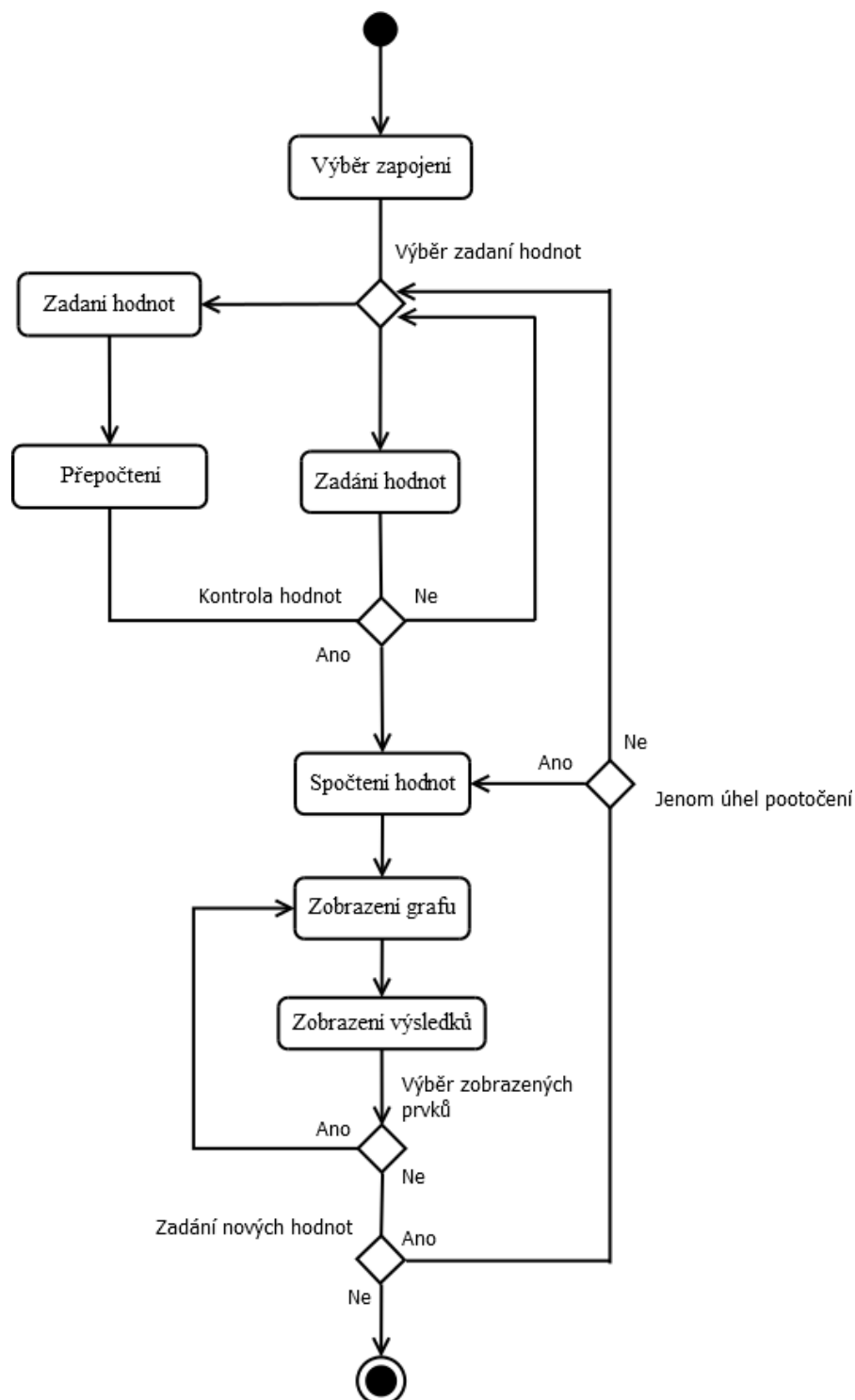
Program pro svou činnost od uživatele očekává, že zadá vstupní parametry jednotlivých prvků. U zdroje napětí je to velikost vstupního napětí, frekvence napětí (frekvence je stejná pro všechny fáze) a jeho vnitřní impedance. Dalším prvkem je vedení, u kterého zadává jeho impedanci. Nakonec je to zátěž, kde se také zadává impedance.

Napětí se u zapojení do hvězdy může zadávat jako fázové, nebo jako sdružené. U zapojení do trojúhelníku se zadává pouze sdružené napětí, jelikož fázové se sdruženému napětí rovná. Impedance se zadává přímo jako komplexní číslo, nebo vyplněním jednotlivých prvků, ze kterých je impedance složena, a tím jsou odpor, cívka a kondenzátor. Nakonec se vybere, pro jaký exponent jsou tyto hodnoty nastaveny (mili, mikro, nano...). Z těchto částí se vypočítá impedance do komplexního tvaru. Komplexní tvar impedance se zadává do dvou polí, kde v prvním poli se zadává reálná část impedance a v druhém poli se zadává imaginární část impedance. Pro desetinná čísla se používá desetinná tečka.

Nakonec si uživatel zvolí, pro které zapojení chce vykreslit diagram a vybere jej (implicitně je nastaveno zapojení do hvězdy – hvězdy se třemi vodiči). Poté stiskne tlačítko Vypočítat a program vypočítá vybranou soustavu zapojení.

Program provede kontrolu, zda jsou zadány všechny požadované hodnoty, a pokud ano, tak následnou kontrolu, jestli jsou zadány ve správném rozmezí a ve správném formátu. Po kontrole se provede výpočet soustavy a nakonec dojde k vykreslení diagramu a otevření okna s výsledky.

4 Implementace programu



Obr. 15 Diagram aktivit

4.1 Programovací jazyk Java

Java je objektově orientovaný programovací jazyk. Tento jazyk jsem zvolil kvůli jeho výhodě nezávislosti na platformě. Tuto výhodu jsem považoval jako hlavní důvod pro využití programu při výuce, kde na univerzitě se velmi často používá systém Linux. Další výhodou tohoto programovacího jazyka je jeho celosvětové rozšíření s možností stáhnout nebo si vytvořit vlastní knihovny.[6]

4.2 Aplikační doména

Aplikační doménu tvoří prvky nesoucí data. Jedná se o třídy `KomplexniCislo` a `Faze`.

4.2.1 Třída `KomplexniCislo`

Při výpočtu napětí a proudů se využívá soustavy rovnic komplexních čísel. Každé komplexní číslo je v programu reprezentováno jako instance třídy `KomplexniCislo`. Třída je doménový objekt a slouží jako nositel dat. Obsahuje pouze dva atributy typu `double`: `realnaCast` a `imaginarniCast`, které odpovídají oběma částem komplexního čísla.

4.2.2 Třída `Faze`

V tomto programu se vypočítává proud a napětí v jednotlivých fázích soustavy. Z tohoto důvodu je zde vytvořena třída `Faze`, která obsahuje proměnné pro jednotlivé počítané prvky, jako jsou napětí, proud, impedance a admitance. Tyto hodnoty jsou typu `KomplexniCislo`. Jelikož se pracuje s trojfázovou soustavou, tak se v tomto programu vytvoří tři instance třídy `Faze`. Tyto instance jsou vytvořeny jako singleton, to znamená, že se vytvoří pouze tyto tři instance a ve všech ostatních třídách bude odkaz právě na tyto instance. To z toho důvodu, že s těmito fázemi se počítá ve více třídách a požaduje se, aby byly hodnoty dané instance dostupné ve všech třídách, a změny v jedné třídě se projeví pro instance ve všech třídách.

4.3 Aplikační logika

Cílem programu je vypočítat hodnoty napětí a proudů na zátěži, vedení a výstupu zdroje jednotlivých fází a jejich proud. Třídy a metody, které tento výpočet realizují, tvoří aplikační logiku programu.

4.3.1 Třída KomplexniCisloKalkulator

Po vytvoření třídy pro reprezentaci komplexních čísel – KomplexniCislo – bylo dále potřeba implementovat základní operace prováděné s komplexními čísly: sčítání, násobení, dělení, odčítání a negaci komplexního čísla. Negaci komplexního čísla se zde rozumí její vynásobení hodnotou -1.

Tyto operace jsou implementovány ve třídě KomplexniCisloKalkulator.

Kód:

```
public KomplexniCislo soucet(KomplexniCislo a, KomplexniCislo b) {  
    double real = a.getRealnaCast() + b.getRealnaCast();  
    double imaginarni = a.getImaginarniCast() + b.getImaginarniCast();  
    return new KomplexniCislo(real, imaginarni);  
}
```

Podle této metody na sečtení dvou komplexních čísel je vidět, že tato operace je podobná pro reálná čísla a to, že se sčítá reálná složka s reálnou a zvlášť imaginární složka s imaginární.

Kód:

```
public KomplexniCislo rozdil(KomplexniCislo a, KomplexniCislo b) {  
    double real = a.getRealnaCast() - b.getRealnaCast();  
    double imaginarni = a.getImaginarniCast() - b.getImaginarniCast();  
    return new KomplexniCislo(real, imaginarni);  
}
```

Metoda na výpočet rozdílu dvou komplexních čísel je také jednoduchá, jelikož pracuje na stejném principu jako metoda pro sčítání dvou komplexních čísel. Zde se však komplexní čísla odčítají.

Kód:

```
public KomplexniCislo soucin(KomplexniCislo a, KomplexniCislo b) {  
    double real = a.getRealnaCast() * b.getRealnaCast() -  
a.getImaginarniCast() * b.getImaginarniCast();  
    double imaginarni = a.getRealnaCast() * b.getImaginarniCast() +  
a.getImaginarniCast() * b.getRealnaCast();  
    return new KomplexniCislo(real, imaginarni);    }
```

Metoda slouží k násobení dvou čísel a zde lze z kódu již vidět, že násobení je složitější. Vztah pro výpočet reálné části je dán tím, že se vynásobí reálné části komplexních čísel a odečte se násobek imaginárních částí komplexních čísel. Imaginární část se vypočte tak, že se reálná část prvního čísla vynásobí s imaginární částí druhého čísla a to se přičte k násobku imaginární části prvního čísla s reálnou částí čísla druhého.

Kód:

```
public KomplexniCislo podil(KomplexniCislo a, KomplexniCislo b) {  
    double delitel = b.getRealnaCast() * b.getRealnaCast() +  
b.getImaginarniCast() * b.getImaginarniCast();  
  
    double real = (a.getRealnaCast() * b.getRealnaCast() +  
a.getImaginarniCast() * b.getImaginarniCast()) / delitel;  
  
    double imaginarni = (a.getImaginarniCast() * b.getRealnaCast() -  
a.getRealnaCast() * b.getImaginarniCast()) / delitel;  
  
    return new KomplexniCislo(real, imaginarni);  
}
```

Tato metoda je opět složitější. Jejím účelem je vydělit dvě čísla. Nejprve se spočte dělitel, kterým se bude dělit jak reálná tak imaginární část. Dělitel se vypočte tak, že u čísla, kterým se dělí, se jeho reálná část na druhou sečte s imaginární částí na druhou. Reálná část výsledného čísla se vypočte stejně jako u násobení reálné části, avšak toto číslo se navíc vydělí dělitelem. Imaginární část je podobná násobení imaginární části, zde se ale vynásobí imaginární část prvního čísla s reálnou částí druhého a odečte se násobek reálné části s imaginární částí čísla druhého. Toto číslo se také ještě vydělí dělitelem spočteným dříve.

Kód:

```
public KomplexniCislo negace(KomplexniCislo k) {  
    return new KomplexniCislo(-1 * k.getRealnaCast(), -1 *  
k.getImaginarniCast()); }  

```

Tato metoda je opět jednoduchá a slouží k změně znaménka u reálné a imaginární části.

4.3.2 Třída ZapojeniKalkulator

Jednotlivá zapojení byla vysvětlena v kapitole (kapitola 2.4.4). Metoda řešící zapojení soustavu hvězda – hvězda se skládá z metod pro výpočet napětí zdroje na komplexní tvar, spočtení impedancí, admitancí, napětí v bodě nula, rozdíl napětí mezi napětím zdroje a bodem nula, výpočtení proudu a nakonec výpočtení jednotlivých napětí.

Tyto metody jsou implementovány ve třídě ZapojeniKalkulator.

Kód:

```
public void napetiZdroje() {  
  
    fazeA.setNapeti(newKomplexniCislo((fazeA.getNapeti().getRealnaCast() *  
Math.cos((0 * Math.PI) / 180)), (fazeA.getNapeti().getRealnaCast() *  
Math.sin((0 * Math.PI) / 180))));  
  
    fazeB.setNapeti(newKomplexniCislo((fazeB.getNapeti().getRealnaCast() *  
Math.cos((240 * Math.PI) / 180)), (fazeB.getNapeti().getRealnaCast() *  
Math.sin((240 * Math.PI) / 180))));  
  
    fazeC.setNapeti(newKomplexniCislo((fazeC.getNapeti().getRealnaCast() *  
Math.cos((120 * Math.PI) / 180)), (fazeC.getNapeti().getRealnaCast() *  
Math.sin((120 * Math.PI) / 180))));  
  
}
```

Zde lze vidět výpočet vztahu (9), který převede zadané napětí na komplexní tvar.

Kód:

```
public void impedanceFaze() {  
  
    fazeA.setCelkovaImpedance(soucet(fazeA.getImpedanceU(), sou-  
cet(fazeA.getImpedanceZ(), fazeA.getImpedanceV())));  
  
    fazeB.setCelkovaImpedance(soucet(fazeB.getImpedanceU(), sou-  
cet(fazeB.getImpedanceZ(), fazeB.getImpedanceV())));  
  
    fazeC.setCelkovaImpedance(soucet(fazeC.getImpedanceU(), sou-  
cet(fazeC.getImpedanceZ(), fazeC.getImpedanceV())));  
  
}
```

Zde lze vidět výpočet vztahu (10), který vypočte celkovou impedanci jedné fáze sečtením impedancí všech prvků fáze.

Kód:

```
public void admittance() {  
  
    KomplexniCislo jedna = new KomplexniCislo(1, 0);  
    fazeA.setCelkovaAdmitance(podil(jedna, fazeA.getCelkovaImpedance()));  
    fazeB.setCelkovaAdmitance(podil(jedna, fazeB.getCelkovaImpedance()));  
    fazeC.setCelkovaAdmitance(podil(jedna, fazeC.getCelkovaImpedance()));  
}
```

Zde lze vidět výpočet vztahu (11), který vypočte impedanci výpočtem obrácené hodnoty impedance. Pro soustavu hvězda – hvězda se čtyřmi vodiči se ještě provede metoda pro vypočtení

admitance pro nulový vodič. Tato metoda je stejná jako předchozí, jenom obsahuje výpočet pouze pro admitanci Y_0 . Pro soustavu hvězda – hvězda se třemi vodiči se nastaví hodnota Y_0 na nulu. Nyní se zavolá metoda pro výpočet napětí v bodě nula.

Kód:

```
public void Uo() {
    KomplexniCislo jmenovatel;
    KomplexniCislo citatel;

    jmenovatel = soucet(soucin(fazeA.getNapeti(), fa-
zeA.getCelkovaAdmitance()), soucet(soucin(fazeB.getNapeti(), fa-
zeB.getCelkovaAdmitance()), soucin(fazeC.getNapeti(), fa-
zeC.getCelkovaAdmitance())));

    citatel = soucet(fazeA.getCelkovaAdmitance(), sou-
cet(fazeB.getCelkovaAdmitance(), soucet(fazeC.getCelkovaAdmitance(),
getYo())));

    setUo(podil(jmenovatel, citatel));
}
```

Zde lze vidět výpočet vztahu (12). Tento výpočet je trochu složitější. Nejprve se vytvoří komplexní čísla pro jmenovatele a čitatele. Dále se vypočte jmenovatel sečtením násobku napětí a admitance dané fáze. Potom se vypočte číťatel tak, že se sečtou všechny admitance. Nakonec se jmenovatel a číťatel podělí a dosadí do proměnné U_0 . Nyní se spočte rozdíl napětí.

Kód:

```
public void rozdílNapeti() {
    fazeA.setNapetiZ(rozdil(fazeA.getNapeti(), uo));
    fazeB.setNapetiZ(rozdil(fazeB.getNapeti(), uo));
    fazeC.setNapetiZ(rozdil(fazeC.getNapeti(), uo));
}
```

Zde lze vidět výpočet vztahu (13). Pomocí odečtení napětí na zdroji od napětí v bodě nula se vypočte napětí mezi bodem nula na zdroji a na zátěži. Výsledky se dosadí mezitím do proměnných pro napětí na zátěži. Dále se vypočtou proudy jednotlivých fází.

Kód:

```
public void proud() {  
    fazeA.setProud(soucin(fazeA.getCelkovaAdmitance(), fazeA.getNapetiZ()));  
    fazeB.setProud(soucin(fazeB.getCelkovaAdmitance(), fazeB.getNapetiZ()));  
    fazeC.setProud(soucin(fazeC.getCelkovaAdmitance(), fazeC.getNapetiZ()));  
}
```

Zde lze vidět výpočet vztahu (14), který se spočte tak, že se násobí admitance fáze a rozdíl napětí mezi napětím zdroje a bodem nula. Nakonec se vypočítá napětí jednotlivých prvků.

Kód:

```
public void vysledneNapeti() {  
    fazeA.setNapetiZ(soucin(fazeA.getImpedanceZ(), fazeA.getProud()));  
    fazeA.setNapetiU(soucin(fazeA.getImpedanceU(), fazeA.getProud()));  
    fazeA.setNapetiV(soucin(fazeA.getImpedanceV(), fazeA.getProud()));  
  
    fazeB.setNapetiZ(soucin(fazeB.getImpedanceZ(), fazeB.getProud()));  
    fazeB.setNapetiU(soucin(fazeB.getImpedanceU(), fazeB.getProud()));  
    fazeB.setNapetiV(soucin(fazeB.getImpedanceV(), fazeB.getProud()));  
  
    fazeC.setNapetiZ(soucin(fazeC.getImpedanceZ(), fazeC.getProud()));  
    fazeC.setNapetiU(soucin(fazeC.getImpedanceU(), fazeC.getProud()));  
    fazeC.setNapetiV(soucin(fazeC.getImpedanceV(), fazeC.getProud()));  
}
```

Zde lze vidět výpočet vztahu (15), tyto napětí se spočítají tak, že se násobí proud fáze a impedance vybraného prvku.

Pro výpočet soustavy hvězda – trojúhelník se zavolá metoda na převod trojúhelníku na hvězdu.

Kód:

```
public void prevodTrojuhelnikHvezda() {  
    KomplexniCislo a, b;  
  
    a = podil(soucin(fazeA.getImpedanceZ(), fazeC.getImpedanceZ()), sou-  
cet(fazeA.getImpedanceZ(), soucet(fazeB.getImpedanceZ(), fa-  
zeC.getImpedanceZ())));  
  
    b = podil(soucin(fazeA.getImpedanceZ(), fazeB.getImpedanceZ()), sou-  
cet(fazeA.getImpedanceZ(), soucet(fazeB.getImpedanceZ(), fa-  
zeC.getImpedanceZ())));  
  
    fazeC.setImpedanceZ(podil(soucin(fazeC.getImpedanceZ(), fa-  
zeB.getImpedanceZ()), soucet(fazeA.getImpedanceZ(), sou-  
cet(fazeB.getImpedanceZ(), fazeC.getImpedanceZ()))));  
  
    fazeA.setImpedanceZ(a);  
    fazeB.setImpedanceZ(b);  
}
```

Zde lze vidět výpočet vztahu (18), po této metodě se zavolá metodu, která se stará o výpočet soustavy hvězda – hvězda se třemi vodiči.

Soustava trojúhelník – trojúhelník se řeší pomocí matice (20). Nejprve se zavolá metoda pro převedení napětí na komplexní tvar a potom se zavolá metoda pro naplnění rovnice daty.

Kód:

```
private void naplneniMatice(Zadani z) {  
    KomplexniCislo jedna = new KomplexniCislo(1, 0);  
    KomplexniCislo a, b, c, d;  
  
    a = podil(jedna, new KomplexniCis-  
lo(Double.parseDouble(z.Zua.getText()),  
Double.parseDouble(z.Zuai.getText())));  
  
    b = podil(jedna, new KomplexniCis-  
lo(Double.parseDouble(z.Zuc.getText()),  
Double.parseDouble(z.Zuci.getText())));  
  
    c = podil(jedna, new KomplexniCis-  
lo(Double.parseDouble(z.Zva.getText()),  
Double.parseDouble(z.Zvai.getText())));  
  
    matice[0][0] = soucet(a, soucet(b, c));  
}
```

```

    matice[0][1] = podil(jedna, new KomplexniCis-
lo(Double.parseDouble(z.Zua.getText()),
Double.parseDouble(z.Zuai.getText())));

    matice[0][1] = negace(matice[0][1]);

        .
        .
        .

    b = podil(fazeC.getNapeti(), new KomplexniCis-
lo(Double.parseDouble(z.Zuc.getText()),
Double.parseDouble(z.Zuci.getText())));

    matice[4][5] = rozdil(a, b);
}

```

Komplexní číslo jedna je děleno všemi čísly, komplexní čísla a, b, c a d jsou zde jako pomocné proměnné pro zpřehlednění kódu. Poté se zavolá metoda, která se stará o převedení matice na jednotkovou matici.

Kód:

```

public void vyreseniMatice() {
    for (int i = 0; i < 4; i++) {
        for (int j = i + 1; j < 5; j++) {
            lTvar(j, i);
        }
    }
    for (int i = 4; i > 0; i--) {
        for (int j = i - 1; j >= 0; j--) {
            luTvar(j, i);
        }
    }
    jednotkovyTvar();
}

```

Metoda lTvar(int a, int b) převede matici na horní trojúhelníkovou matici, kde parametr a představuje upravovaný řádek a parametr b představuje řádek, pod kterým se mají nacházet nuly.

Kód:

```
public void lTvar(int a, int b) {  
    KomplexniCislo cislo;  
    cislo = podil(matice[a][b], matice[b][b]);  
    matice[a][b] = new KomplexniCislo(0, 0);  
    for (int j = b + 1; j < 6; j++) {  
        matice[a][j] = rozdil(matice[a][j], soucin(matice[b][j], cislo));  
    }  
}
```

Po této metodě se zavolá `luTvar(int a, int b)`, která převede horní trojúhelníkovou matici na diagonální matici, kde budou čísla jenom na diagonále. Zde představuje parametr `a` upravovaný řádek a parametr `b` představuje řádek, nad kterým se mají nacházet nuly.

Kód:

```
public void luTvar(int a, int b) {  
    KomplexniCislo cislo;  
    cislo = podil(matice[a][b], matice[b][b]);  
    matice[a][5] = rozdil(matice[a][5], soucin(matice[b][5], cislo));  
    matice[a][b] = new KomplexniCislo(0, 0); }  
}
```

Nakonec se zavolá metoda `jednotkovyTvar`, která převede diagonální matici na její jednotkový tvar, a tím se získají daná napětí.

Kód:

```
private void jednotkovyTvar() {  
    for (int i = 0; i < 5; i++) {  
        matice[i][5] = podil(matice[i][5], matice[i][i]);  
        matice[i][i] = podil(matice[i][i], matice[i][i]);  
    }  
}
```

Po výpočtu daných napětí, se vypočítají zbývající pomocí zjištěných napětí.

Kód:

```
public void reseniDD() {  
    fazeA.setNapetiV(matice[0][5]);  
    fazeB.setNapetiV(matice[1][5]);  
    fazeC.setNapetiV(matice[2][5]);  
    fazeA.setNapetiZ(matice[3][5]);  
    fazeB.setNapetiZ(matice[4][5]);  
    fazeC.setNapetiZ(negace(soucet(fazeA.getNapetiZ(), fazeB.getNapetiZ())));  
    fazeA.setNapetiU(rozdil(fazeA.getNapeti(), (soucet(fazeA.getNapetiV(), rozdil(fazeA.getNapetiZ(), fazeB.getNapetiV())))));  
    fazeB.setNapetiU(rozdil(fazeB.getNapeti(), (soucet(fazeB.getNapetiV(), rozdil(fazeB.getNapetiZ(), fazeC.getNapetiV())))));  
    fazeC.setNapetiU(rozdil(fazeC.getNapeti(), rozdil(fazeC.getNapetiV(), soucet(fazeA.getNapetiV(), soucet(fazeA.getNapetiZ(), fazeB.getNapetiZ())))));  
    uo = new KomplexniCislo(0, 0);  
}
```

Tímto je vypočtena soustava zapojená do trojúhelníku – trojúhelníku.

Pro spočtení soustavy do trojúhelníku – hvězdy se nejprve převede hvězda na trojúhelník a dále se zavolá metoda, která řeší soustavu trojúhelník – trojúhelník.

Kód:

```
public void prevodHvezdaTrojuhelnik() {  
    KomplexniCislo a, b, jmenovatel;  
    jmenovatel = soucet(soucin(fazeA.getImpedanceZ(), fazeB.getImpedanceZ()), soucet(soucin(fazeA.getImpedanceZ(), fazeC.getImpedanceZ()), soucin(fazeB.getImpedanceZ(), fazeC.getImpedanceZ())));  
    a = podil(jmenovatel, fazeC.getImpedanceZ());  
  
    jmenovatel = soucet(soucin(fazeA.getImpedanceZ(), fazeB.getImpedanceZ()), soucet(soucin(fazeA.getImpedanceZ(), fazeC.getImpedanceZ()), soucin(fazeB.getImpedanceZ(), fazeC.getImpedanceZ())));
```

```

        b = podil(jmenovatel, fazeA.getImpedanceZ());

        jmenovatel = soucet(soucin(fazeA.getImpedanceZ(), fa-
zeB.getImpedanceZ()), soucet(soucin(fazeA.getImpedanceZ(), fa-
zeC.getImpedanceZ()), soucin(fazeB.getImpedanceZ(), fa-
zeC.getImpedanceZ())));

        fazeC.setImpedanceZ(podil(jmenovatel, fazeB.getImpedanceZ()));

        fazeA.setImpedanceZ(a);

        fazeB.setImpedanceZ(b);

    }

```

Nyní je vypočtena vybraná soustava a vykreslí se diagram.

4.4 Grafické rozhraní

V tomto programu se bude používat knihovna Swing. Tato knihovna slouží k interakci grafického rozhraní. Pomocí této knihovny lze vytvářet okna, tlačítka, výběry a další. Z této knihovny se využije `JFrame`, jako okno, dále `JPanel` jako kontejnery, které budou obsahovat nastavení programu, `JTextField` sloužící jako vstupní pole, `JLabel` sloužící k popisu zadávaných hodnot do `JTextField`, dále `JComboBox` sloužící k výběru zadávání hodnot, `JCheckBox` pro výběr vykreslení do diagramu, `JSpinner` sloužící pro nastavení úhlu o jaký se diagram pootočí a `JBButton` pro vyvolání a zpracování událostí.

Grafické rozhraní bylo vytvořeno tak, že se ve vývojovém prostředí NetBeans zvolil design. Zde se může grafické rozhraní tzv. “naklikat“. Poté co se toto rozhraní “naklikalo“, tak NetBeans vytvoří automaticky zdrojový kód pro zobrazení, ale nikoliv pro funkčnost.

4.4.1 Implementace vykreslení diagramu

Nejprve se vytvoří instance třídy Vykreslení, kde jsou parametry `JPanel`, do kterého se bude kreslit fázorový diagram, hodnoty napětí v bodě nula U_0 a konkrétní třída zadání. Nakreslí se plný obdélník, aby se smazal předchozí diagram. Poté se vykreslí pozadí diagramu (osy x a y, stupně atd.). Dále se zjistí největší napětí a z něj se vypočítá převod. Ten slouží k informaci o tom, jak velkou hodnotu napětí představuje jeden pixel, aby největší napětí bylo na kružnici, která se rovná 100% napětí. Dále se vypočítá z komplexního čísla jeho úhel. Poté se nastaví vlastnosti čáry (plná, přerušovaná, tečkovaná, atd.). Nakonec se zavolá metoda, která vypočte souřadnice bodu x a y podle velikosti komplexního čísla, a dále se z této metody zavolá metoda, která z nastaveného středu vykreslí do tohoto bodu fázor. Podle napětí, které se vykresluje, se mění vzhled šipky. Střed se posune po vypočtení a vykreslení fázoru pro hodnotu napětí v bodě nula U_0 . Toto posunutí středu platí pouze pro napětí nacházející se na zátěži, proto se napětí na zátěži kreslí jako poslední, až po napětí U_0 . Po tom, co se vykreslí jednotlivé fázory napětí, se začnou vykreslovat fázory proudu. Nejprve se opět vypočte převod, tentokrát pro proud. Dále se

zavolá metoda pro výpočet proudu a vykreslení fázoru, avšak tentokrát s vyplněnou šipkou. Nakonec se nakreslí fázory pro sdružené napětí.

Poté, co se vykreslí diagram, se může pomocí `JSpinneru` pootáčet diagram pomocí zvětšení nebo zmenšení úhlu.

4.4.2 Implementace zobrazení výsledků

Výsledky se zobrazí v novém okně. Nejprve se vytvoří instance třídy `Vysledky`, kde parametry konstruktoru jsou typ zapojení soustavy, napětí v bodě nula U_0 a admitance v bodě nula Y_0 . V tomto novém okně se nachází `JPanel`, který obsahuje `JLabely`, ve kterých se budou měnit výsledky na text. Nejprve se zavolá metoda pro nastavení textu `JLabelu`. Tato metoda bude mít parametr textový řetězec, jenž bude obsahovat znak proměnné například U_0 pro napětí v bodě nula. Dále znak rovná se a hodnotu komplexního čísla. Pro hodnotu komplexního čísla se zavolá metoda pro jeho výpis. Tato metoda nejprve upraví reálnou a imaginární část tak, aby se zobrazovala pro přehlednost pouze dvě desetinná čísla. Poté se zjistí velikost imaginární části pro informaci, jestli napsat znaménko plus nebo minus a tímto metoda končí. Dále se pokračuje opět znaménkem rovná se a další metodou, která slouží k převodu čísla z komplexního tvaru na tvar goniometrický. Tato metoda má parametry komplexní číslo a znak jednotky pro danou veličinu. Opět je zde úprava formátu na dvě desetinná čísla a dále metoda pro zjištění úhlu. Nakonec se převede komplexní tvar na goniometrický, a přidá znak jednotky, dále se napíše znak představující úhel a jeho velikost a nakonec znak pro stupeň. Toto se provede pro všechna napětí, proudy a admitance.

4.4.3 Interakce s uživatelem

První událostí je zmačknutí tlačítka pro výběr zapojení. Z ukázky kódu lze vidět, že po stlačení tlačítka Výběr zapojení se otevře okno pro třídu `Zapoj`, která se stará o typ zapojení.

Kód:

```
private void vyberZapojeniActionPerformed(java.awt.event.ActionEvent
evt) {
    zapoj.setVisible(true);
}
```

V další části kódu lze vidět zpracování výběru typu zapojení, kde se nejprve zakáže možnost editace vstupního pole pro zadávání hodnot impedance vedení nulového vodiče Z_0 . Poté se zjišťuje, které zapojení bylo vybráno z rozbalovací seznamu typů zapojení. Jakmile se najde typ

zapojení, změní se hodnota proměnné `typZapojeni` na hodnotu vybraného zapojení, obrázek `JLabelu` `schema` se změní na obrázek představující toto zapojení a nakonec se změní text `JLabelu` `zapojeniInfo` na text informující jaké zobrazení je vybráno. V případě, že je vybráno zapojení typu hvězda – hvězda se čtyřmi vodiči, tak se navíc povolí zadávání hodnot impedance pro vedení nulového vodiče Z_0 .

Kód:

```
private void zapojeniActionPerformed(java.awt.event.ActionEvent evt) {
    zadani.Zvo.setEditable(false);
    zadani.Zvoi.setEditable(false);
    if (zapojeni.getSelectedIndex() == 0) {
        zadani.setTypZapojeni(0);
        schema.setIcon(new ImageIcon(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/obrazky/YY3.png"))));
        zadani.zapojeniInfo.setText("Hvězda - Hvězda - YY3");
    }
    if (zapojeni.getSelectedIndex() == 1) {
        zadani.Zvo.setEditable(true);
        zadani.Zvoi.setEditable(true);
        zadani.setTypZapojeni(1);
        schema.setIcon(new ImageIcon(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/obrazky/YY4.png"))));
        zadani.zapojeniInfo.setText("Hvězda - Hvězda - YY4");
    }
    .
    .
    .
    if (zapojeni.getSelectedIndex() == 4) {
        zadani.setTypZapojeni(4);
        schema.setIcon(new ImageIcon(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/obrazky/DD.png"))));
    }
}
```

```

        zadani.zapojeniInfo.setText(zapojeni.getSelectedItem().toString());
    }
}

```

Nakonec po kliknutí na tlačítko OK se provede metoda, která toto okno uzavře:

Kód:

```

private void potvrzeniActionPerformed(java.awt.event.ActionEvent evt)
{
    this.dispose();
}

```

Další událostí je výběr typu zadávání hodnot, toto se provede zmáčknutím příslušného přepínače. U možností zapojení se po výběru zavolá metoda, která vybere, zda se budou zadávat hodnoty fázového nebo sdruženého napětí.

Kód

```

private void jRadioNapetiUsActionPerformed(java.awt.event.ActionEvent evt) {
    Ua.setEditable(false);
    Ub.setEditable(false);
    Uc.setEditable(false);
    Us.setEditable(true);
}

```

Tato metoda umožní zadávání sdruženého napětí. Metoda pro zadávání fázového napětí je stejná, liší se jenom v boolean hodnotách, které jsou opačně (místo false je true a naopak). Další výběr typu zadávání, tentokrát impedance se provede pomocí metody, funguje na stejném principu, avšak kromě vstupních polí navíc povoluje a zakazuje používání tlačítek na otevření okna pro zadávání jednotlivých prvků impedance.

Třetí událostí je zadání prvků a její následující výpočet. Pro zadávání ústředního napětí se použije metoda pro přepočítání ústředního napětí na fázové:

Kód:

```
private void prepocet(java.awt.event.FocusEvent evt) {  
    DecimalFormat df = new DecimalFormat("#.##");  
    String novaHodnota = "" +  
df.format((Double.parseDouble(Us.getText()) / Math.sqrt(3)));  
    novaHodnota = novaHodnota.replace(',', '.', '.');  
    Ua.setText(novaHodnota);  
    Ub.setText(novaHodnota);  
    Uc.setText(novaHodnota);  
}
```

Zde se nejprve vytvoří formát pro hodnotu `Double`, který zařídí to, že `Double` bude mít pouze dvě desetinná místa, a to proto, aby přepočtené číslo bylo lépe čitelné v zadání, kde se nakonec dosadí do vstupních polí pro fázové napětí. Poté se převede sdružené napětí na fázové pomocí vztahu (5). Tento přepočet se použije z důvodu, že program počítá pouze s fázovým napětím.

Další zadávání se nachází u impedance, kde se po otevření okna a zadání prvků provede výpočet impedance pro dané prvky. Nejprve se zjistí, jakého řádu je mocnina a dosadí se hodnota do proměnné pomocí metody.

Kód:

```
switch (MocninaR.getSelectedIndex()) {  
    case 0:  
        setMocninaR(Math.pow(10, 9));  
        break;  
        .  
        .  
        .  
    case 9:  
        setMocninaR(Math.pow(10, -18));  
        break;  
}
```

Toto se provede pro všechny hodnoty řádu mocnin. Poté se použije metoda, ve které se nastaví hodnoty pro jednotlivé fáze, a zavolá se metoda, která se postará o výpočet impedance. Nakonec se dosadí dané výsledky do textových polí pro zadávání impedance pro případ, že by se měly tyto hodnoty změnit.

Kód:

```
try {
    if (misto == 0) {
        fazeA.setImpedanceZ(new KomplexniCislo(Double.parseDouble(Ra.getText()), 0));
        fazeB.setImpedanceZ(new KomplexniCislo(Double.parseDouble(Rb.getText()), 0));
        fazeC.setImpedanceZ(new KomplexniCislo(Double.parseDouble(Rc.getText()), 0));
        fazeA.setLz(Integer.parseInt(La.getText()));
        fazeB.setLz(Integer.parseInt(Lb.getText()));
        fazeC.setLz(Integer.parseInt(Lc.getText()));
        fazeA.setCz(Integer.parseInt(Ca.getText()));
        fazeB.setCz(Integer.parseInt(Cb.getText()));
        fazeC.setCz(Integer.parseInt(Cc.getText()));

        spocti.spocetniImpedance(Integer.parseInt(zadani.Frekvence.getText()),
            mocninaL, mocninaC, misto);

        zadani.Zza.setText((int) fazeA.getImpedanceZ().getRealnaCast() +
            "");
        zadani.Zzai.setText((int) fazeA.getImpedanceZ().getImaginarniCast()
            + "");
        zadani.Zzb.setText((int) fazeB.getImpedanceZ().getRealnaCast() +
            "");
        zadani.Zzbi.setText((int) fazeB.getImpedanceZ().getImaginarniCast()
            + "");
        zadani.Zzc.setText((int) fazeC.getImpedanceZ().getRealnaCast() +
            "");
        zadani.Zzci.setText((int) fazeC.getImpedanceZ().getImaginarniCast()
            + "");
    }
}
```

```

        .
        .
        .

        this.dispose();
    }
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(jButton1, "Špatně zadaná čísla.\nProsím vyplňte všechny požadované parametry.\n(Pro desetinné číslo použijte tečku.)");
}

```

Proměnná *misto* představuje pro jaké místo se počítají dané hodnoty, tato místa jsou zátěž, zdroj nebo vedení. Místo se určí podle zmačknutého tlačítka pro dané místo. Nejprve se vypočte hodnota pro induktivní reaktanci.

Kód:

```

public void indukcnost(int frek, double mocnina, int misto) {
    if (misto == 0) {
        fazeA.setLz((int) (2 * Math.PI * frek * fazeA.getLz() * mocnina));
        fazeB.setLz((int) (2 * Math.PI * frek * fazeB.getLz() * mocnina));
        fazeC.setLz((int) (2 * Math.PI * frek * fazeC.getLz() * mocnina));
    }
    .
    .
    .
}

```

Zde se provede výpočet pro vybrané místo, které se předá pomocí parametru metody. Poté se spočítá kapacitní reaktance.

Kód:

```

public void kapacita(int frek, double mocnina, int misto) {
    if (misto == 0) {
        fazeA.setCz(-(1 / (2 * Math.PI * frek * fazeA.getCz() * mocnina)));
        fazeB.setCz(-(1 / (2 * Math.PI * frek * fa-

```

```

zeB.getCz() * mocnina)));
        fazeC.setCz(-(1 / (2 * Math.PI * frek * fa-
zeC.getCz() * mocnina)));
    }

        .
        .
        .
}

```

Zde se provede výpočet pro vybrané místo, které se předá pomocí parametru metody. A na-
konec se spočte imaginární část impedance, kde se opět se provede výpočet pro vybrané místo,
které se zde předá pomocí parametru metody.

Kód:

```

if (misto == 0) {
    fazeA.setImpedanceZ(new KomplexniCis-
lo(fazeA.getImpedanceZ().getRealnaCast(), fazeA.getLz() + fa-
zeA.getCz()));
    fazeB.setImpedanceZ(new KomplexniCis-
lo(fazeB.getImpedanceZ().getRealnaCast(), fazeB.getLz() + fa-
zeB.getCz()));
    fazeC.setImpedanceZ(new KomplexniCis-
lo(fazeC.getImpedanceZ().getRealnaCast(), fazeC.getLz() + fa-
zeC.getCz()));
}

        .
        .
        .
}

```

Nyní uživatel může pracovat s programem.

5 Závěr

Vytvořený program slouží pro zobrazování fázorových diagramů přenosové sítě na počítači. Propočítává proudy a napětí na jednotlivých prvcích fází a následně vykresluje fázorový diagram a zobrazuje okno obsahující vypočtené výsledky. Je umožněn výběr několika soustav zapojení a výběr impedancí, které se buď zadají, nebo se budou rovnat nule a vytvoří se tak ideální zdroj nebo vedení. Zobrazovaný diagram může být celkově pootočený o určitý úhel. Uživatel má také možnost vybrat, které prvky se mají na diagramu zobrazovat.

Na zpracování programu byla nejnáročnější část zpracování matematické části programu. Po provedených výpočtech dochází k vykreslení diagramu, což se provádí implementovanými metodami. Grafické rozhraní bylo vytvořeno pomocí knihovny Swing ve vývojovém prostředí NetBeans.

Pro vytvoření programu byl použit programovací jazyk Java. Především kvůli jeho výhodě nezávislosti na platformě z důvodu často používaného systému Linux.

Program bude sloužit především jako podpůrný prostředek ve výuce zabývající se řešením trojfázových obvodů. Například v předmětech Teorie elektrických obvodů nebo elektrická měření.

Vytvořením funkčního programu pro zobrazování fázorového diagramu přenosových sítí na počítači bylo dosaženo stanoveného cíle této práce.

Dále může být program rozšířen o nové možnosti. K těmto možnostem patří například přidání soustav se zapojením zdrojů a zátěží do lomené hvězdy a jejich kombinace, dále přidáním prvků jako například přidáním transformátoru do vedení, který bude sloužit k úpravě napětí a proudu.

6 Literatura

- [1] Székely J., *Teoretická elektrotechnika I*, 7. vydání, ALFA, Bratislava, 1968.
- [2] Kijonka J., *Teorie obvodů II*, 1. vydání, VŠB – TUO, Ostrava, 2007.
- [3] Kratochvíl F., Trojfázové obvody [online].
http://home.zcu.cz/~karban/teaching/te1/3f/kratochvil_3f.pdf.
- [4] Székely J., *Příklady z teoretické elektrotechniky*, ALFA, Bratislava, 1966.
- [5] Mohylová J., *Cvičení z Elektrických obvodů I*, 1. vydání, VŠB – TUO, Ostrava, 2007.
- [6] Java technologie [online], Oracle,
<http://www.oracle.com/us/technologies/java/overview/index.html>.

7 Přílohy

Obsah CD

- Samostatná práce ve formátu .pdf
- Zdrojové kódy ve formě projektu do NetBeans IDE
- Program
- Instalace programu
- Java runtime environment
- Manuál pro instalaci a práci s programem